IJ Instruments Ltd.

# IJ-6 SOFTWARE-BASED TEMPERATURE CONTROLLER

# USER MANUAL AND REFERENCE

MANUAL VERSION 1.3

FOR SOFTWARE VERSION 1.3 AND HARDWARE REVISION C

# 1 TABLE OF CONTENTS


2 Introduction & Specifications ..... 4

3 Safety ..... 4

4 FCC Notice ..... 5

5 IJ-7 Demo Module ..... 5

6 Installation and Configuration ..... 5

6.1 Software Installation ..... 5

6.1.1 Install Serial Port Driver ..... 5

6.1.2 Configure Port ..... 6

6.1.3 Install Control Software ..... 7

6.2 Software Configuration ..... 7

6.2.1 Running the Control Software ..... 7

6.2.2 How to Enter Data ..... 8

6.2.3 ADC Configuration ..... 9

6.2.4 Moving the Window ..... 9

6.3 Ready to Use ..... 9

6.4 Uninstallation ..... 9

7 PID Theory ..... 9

7.1 PID Equation ..... 10

7.1.1 Proportional (P) Control ..... 11

7.1.2 Proportional+Integral (PI) Control ..... 12

7.1.3 Proportional+Integral+Derivative (PID) Control ..... 12

7.2 Tuning ..... 13

7.2.1 Closed-Loop ..... 13

7.2.2 Open-Loop ..... 14

7.2.3 Summary + Table of Tuning Factors ..... 17

8 Hardware Reference ..... 18

8.1 Rear Panel ..... 18

8.1.1 USB Connection ..... 18

8.1.2 TXD/RXD Indicators ..... 18
8.1.3 Power Switch ..... 18
8.2 Front Panel ..... 18
8.2.1 General-Purpose Relay Outputs ..... 18
8.2.2 Main Solid-State Relay Output ..... 19
8.2.3 Thermocouple Ground Reference ..... 20
8.2.4 Status Indicators ..... 22
8.2.5 Thermocouple Input ..... 22
9 Software Reference ..... 22
9.1 Power Control ..... 22
9.2 ADC Settings ..... 23
9.3 Current Value Display ..... 24
9.4 Thermocouple Type ..... 24
9.5 Logging ..... 24
9.6 PID Parameters and Gain Scheduling ..... 26
9.7 Setpoint and Program Control ..... 27
9.8 Relay Control ..... 29
9.9 Graphical Display ..... 30
10 Calibration ..... 30
10.1 ADC Zero ..... 31
10.2 ADC Gain ..... 31
11 Miscellaneous ..... 31
11.1 Disconnecting the Controller ..... 31
11.2 Maximum Recording Time Limit ..... 31
11.3 Heavy CPU Usage ..... 32
11.4 Operating Temperature Variations ..... 32
12 Product Disposal Instructions ..... 33
13 Contact/Support ..... 34
14 Revision History ..... 34

## 2 INTRODUCTION & SPECIFICATIONS

The IJ-6 temperature controller provides a means of controlling many temperature processes through a computer-based interface. Unlike many PID controllers which are self-contained and can be inconvenient to program, the IJ-controller is entirely software-based, giving greater flexibility and ease of use. Log files, temperature programs, gain schedules and a graphical display are all readily available through the software.

**Specifications**

| | |
|---|---|
| Power: | Self-powered USB |
| Input connector: | Miniature thermocouple socket |
| Input voltage range: | -30mV to +80mV |
| Thermocouple type: | E, J, K, N, T (limited only by software) |
| Voltage resolution: | 2µV |
| Temperature resolution: | Type E 0.03°C |
| | Type J 0.04°C |
| | Type K 0.05°C |
| | Type N 0.06°C |
| | Type T 0.04°C |
| | (These are the temperatures equivalent to the voltage resolution. Actual resolution and accuracy will be limited by the choice of thermocouple and may be greater.) |
| Ambient temperature resolution: | 0.03°C |
| Outputs: | Variable duty cycle SSR drive (11V 40mA max) |
| | Two general-purpose SSR outputs (50V 0.2A max) |

**Software features**

Display of process, ambient and setpoint temperatures
File logging of all variables
Graphical temperature display
Programmable temperature profiles
Gain scheduling of PID parameters

## 3 SAFETY

Some form of isolation (e.g. relay or optoisolator) **must always** be used between the controller and the controlled appliance. The **ONLY** exception to this rule is when using the controller with the IJ-7 Demonstration Module, since the IJ-7 has been designed for direct connection to the controller.

Additional protection **must be used**, for example a safety thermostat, to protect the controlled system against overheating in the event of failure of the temperature controller.

As a safety measure, if the controller does not receive any communication from the host computer within approximately 7 seconds (e.g. due to a software failure), it turns all of its outputs off (the solid state relay output and both general-purpose relay outputs). This ensures that any connected equipment is automatically switched off.

Although the controller is immune to normal levels of electromagnetic interference, such as might be experienced in a laboratory environment, extremely high levels may cause unpredictable operation. Reasonable measures should be taken to reduce the impact of noise on the controller.

The two general-purpose relay outputs (Section 8.2.1) are floating and can be damaged by electrostatic discharge from personnel (e.g. by wearing woollen clothing or walking across a carpet). If in doubt, please ground yourself first, by touching a grounded metal object, before working with the front-panel terminal block.

IJ Instruments Ltd. accepts no responsibility for loss or damage to property caused by careless or incorrect usage of this device.

# 4 FCC NOTICE

This device complies with part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) This device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures: —Reorient or relocate the receiving antenna. —Increase the separation between the equipment and receiver. —Connect the equipment into an outlet on a circuit different from that to which the receiver is connected. —Consult the dealer or an experienced radio/TV technician for help.

# 5 IJ-7 DEMO MODULE

If the user is new to PID control, they may find it helpful to purchase the IJ-7 demo module which is intended for use with the IJ-6 controller. This demo module contains a small aluminium block which is heated with a power resistor. The temperature of the block can be measured with the supplied bead thermocouple. The module also has a buzzer which can be switched by one of the general-purpose relay outputs on the IJ-6 controller. These features allow the user to explore the operation of the IJ-6 controller.

This manual is written with operation of the IJ-7 demo module in mind (to allow the user to reproduce any of the graphs shown), but the underlying principles are applicable to any temperature control system.

# 6 INSTALLATION AND CONFIGURATION

Follow the steps given in these sections to install and configure the software for first use. Ensure that the controller is initially disconnected from the computer and switched off. Please note: when installing a new version of the software, manually uninstall the older version first.

The software has been tested on Windows XP, 7, and 8.

## 6.1 SOFTWARE INSTALLATION

### 6.1.1 INSTALL SERIAL PORT DRIVER

The controller communicates with the host computer through a virtual COM serial port, provided by FTDI (http://www.ftdichip.com). A driver installation program for both 32-bit and 64-bit versions of Windows is

available on the enclosed CD, from the "Downloads" page of the IJ Instruments' website at http://www.eyejayinstruments.com, or from FTDI's driver page at http://www.ftdichip.com/Drivers/VCP.htm. If downloading from FTDI's site, choose the link under "Available as setup executable". The installation program is called CDM*****_Setup.exe, where the asterisks refer to the version number.

Run the program (it may have to be double-clicked *twice*) and driver installation should proceed automatically. Accept any Windows messages which may appear.

Once the driver has been installed, connect the IJ-6 controller using the supplied cable and switch on by pressing the switch at the rear of the unit. The "Found New Hardware" message may appear – wait for this to finish and say "Your new hardware is installed and ready to use" or a similar message. The controller has now been recognised by the computer.

### 6.1.2 CONFIGURE PORT

With the controller connected, go in to Device Manager. This can be accessed most quickly by entering **devmgmt.msc** into Start Menu -> Run (Windows XP) or Start Menu -> Search box (Windows Vista and 7) and pressing Enter. In Windows 8, enter **devmgmt.msc** into the search box on the Start screen.

Under "Ports (COM & LPT)", double-click on the port corresponding to the temperature controller to bring up its properties. If unsure which port to choose, switch the controller off and see which port disappears, and then switch it back on. Go to Port Settings -> Advanced and you should see a window similar to that in Fig. 1.

From the drop-down list to the right of "Latency Timer (msec):", select 1. Ignore all the other settings. Click OK and close the various windows which are still open. The port is now properly configured.

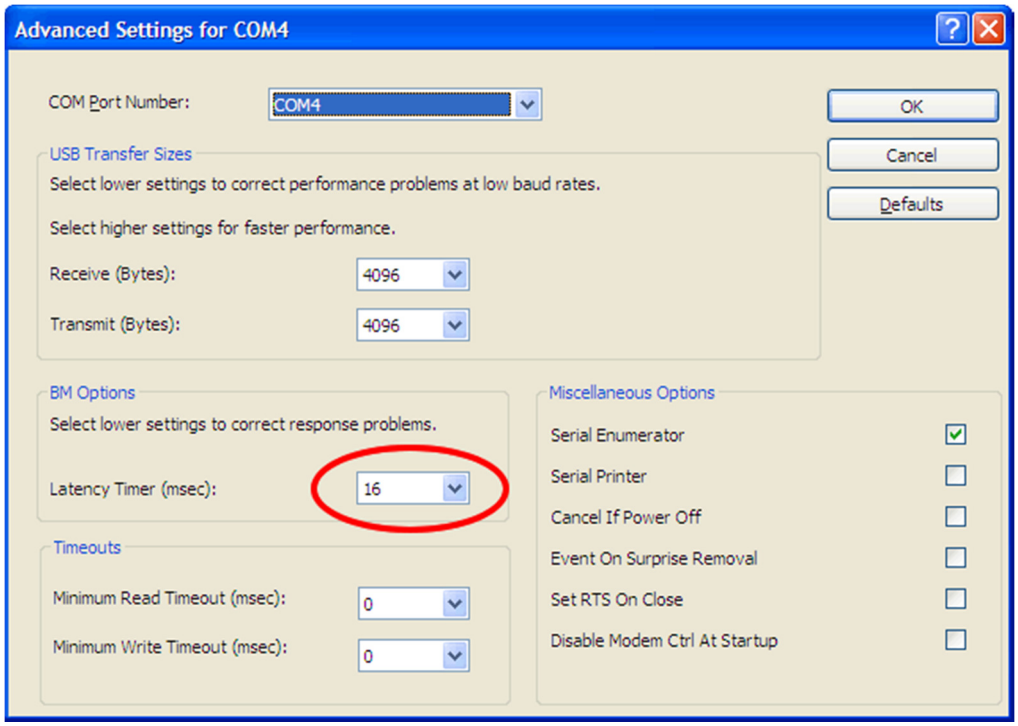



**Fig. 1: Port settings dialog**

### 6.1.3 INSTALL CONTROL SOFTWARE

Next, install the temperature control software. An installer (called “Temperature control software v1.3 setup.exe” or similar) is available either on the enclosed CD or from the “Downloads” section at http://www.eyejayinstruments.com.

Run the installer program. Again, accept any Windows messages which may appear. Once installed, an entry will appear under Start Menu -> Programs as usual.

## 6.2 SOFTWARE CONFIGURATION

### 6.2.1 RUNNING THE CONTROL SOFTWARE

When first run, the user is presented with the splash screen shown in Fig. 2. Ensure the controller is connected and powered on, then click “Search for devices”. If the controller is found, the associated COM port and the controller’s serial number will be shown in the list. If no controller is found, try switching the controller off and on and searching again.

Click on the device to use and click “Use selected”. The splash screen will disappear and the main control screen appears (Fig. 3). The device’s serial number is shown in the window title bar. The program automatically reloads the previous settings (if used), saving the user having to reconfigure the program every time it is run.

The values displayed, both on the graph and in the indicators, are updated approximately every second, so there may be a small delay before the program responds to user input.

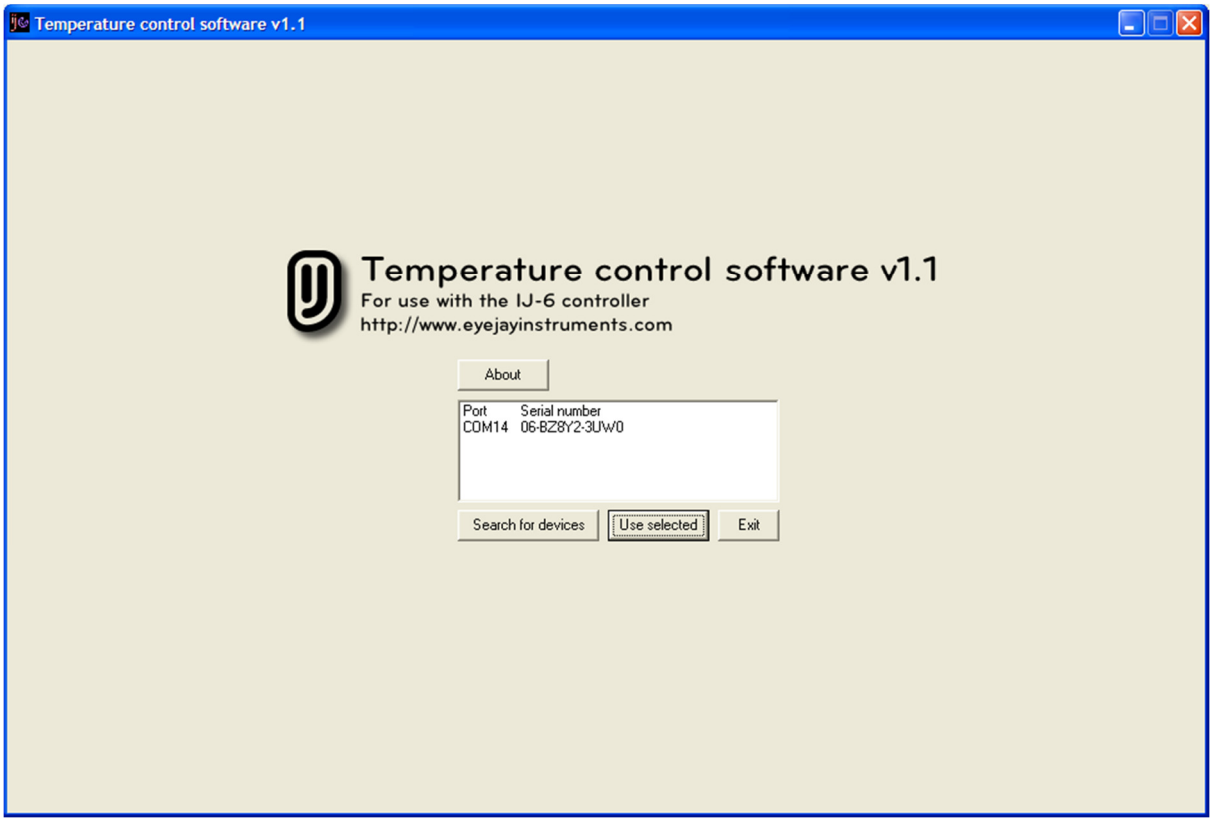



**Fig. 2: Splash screen**

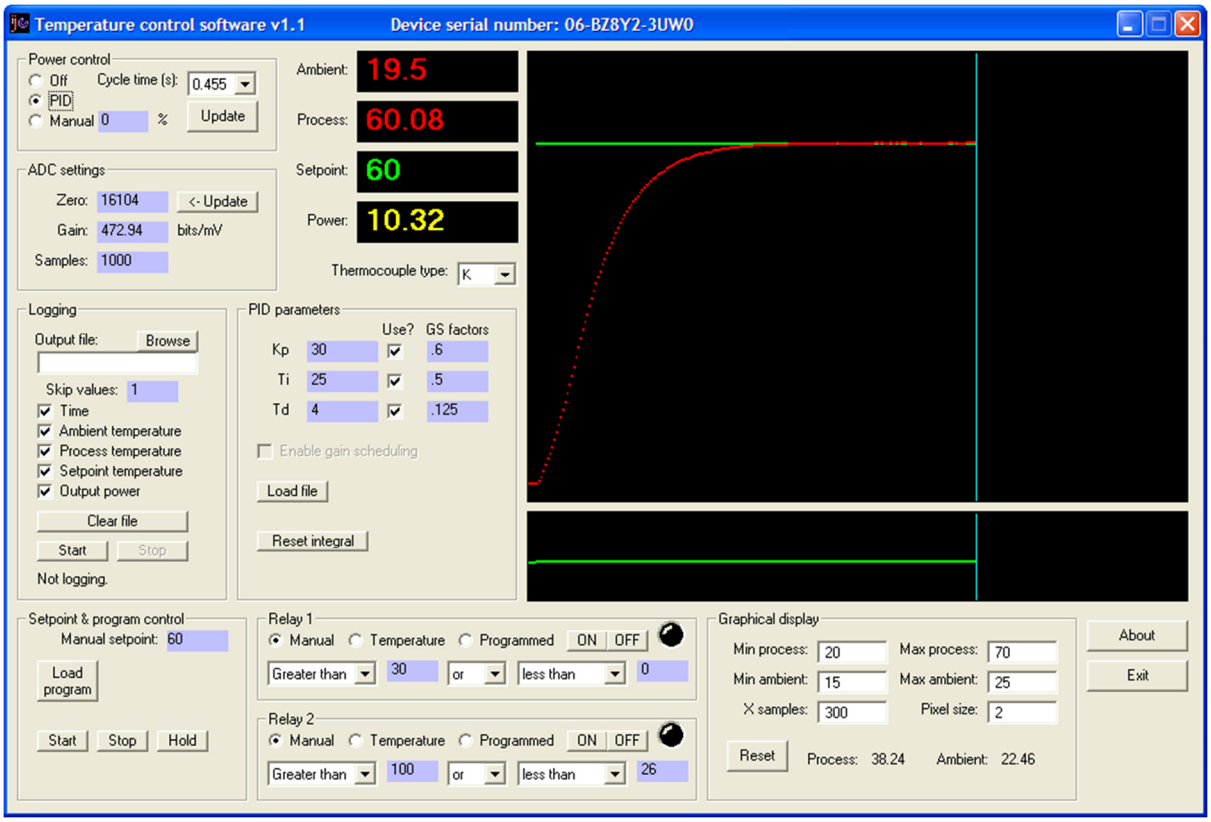



**Fig. 3: Main screen (shown after recording some temperature data)**

## 6.2.2 HOW TO ENTER DATA

Several parameters in the program are entered in the blue-coloured boxes. The behaviour of these is slightly different from a normal entry box. The process is shown in Fig. 4.

Click on the value to be changed. The box will turn red and you can enter a new numerical value in the red box as you would normally. To accept the new value, press Return or Enter when the cursor is in the box. The box turns back to blue, with the new value present. If you change your mind and don't want to accept the new value, press the Escape key when the cursor is in the red box. It will leave the value in the blue box unchanged.

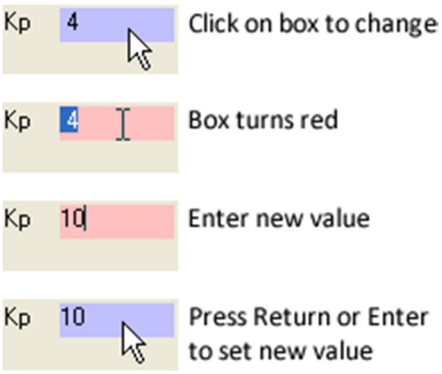



**Fig. 4: Entering data**

### 6.2.3 ADC CONFIGURATION

The controller is provided with a calibration sheet which lists values for “Zero” and “Gain”. These values are slightly different for every controller and **must** be entered in the software to ensure accurate operation. For an explanation of these values, please refer to Section 9.2.

Enter the zero and gain values from the calibration sheet into the corresponding boxes in the “ADC settings” section of the program window, as shown in Fig. 5. Ignore the other boxes and buttons for the moment.

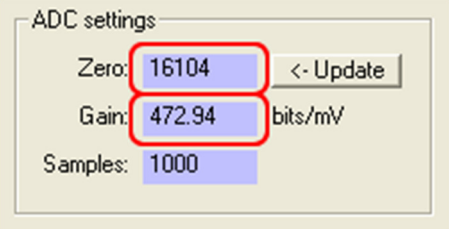



**Fig. 5: Entering calibration data**

### 6.2.4 MOVING THE WINDOW

If the user moves the program window by clicking and dragging on the title bar, it will sometimes temporarily pause to allow the window to be moved. If the window is moved for more than 7 seconds, the controller itself can timeout (see Section 3) as a safety feature. To prevent this, do not move the window for longer than 7 seconds.

## 6.3 READY TO USE

At this stage, the software is ready to use. If desired, please refer now to Section 7 for PID Theory, Section 8 for a detailed hardware reference, and Section 9 for a full software reference. If you have the IJ-7 demo module, then you may now proceed with the experiments described in its manual.

## 6.4 UNINSTALLATION

The temperature control software can be uninstalled as normal through Control Panel -> Add/Remove Programs (Windows XP) or Control Panel -> Programs and Features (Windows Vista and 7).

Because the virtual serial port is a driver, it will not appear in Add/Remove programs. If it is necessary to uninstall the FTDI driver, this can be done using the “CDM Uninstaller 1.4 - Windows Device Driver Uninstaller” utility available from FTDI at http://www.ftdichip.com/Support/Utilities.htm. Please refer to FTDI’s documentation for details of the process.

# 7 PID THEORY

The benefits of PID control are demonstrated in Fig. 6, which compares the results of using either on/off or PID control to regulate the temperature of the IJ-7 module to a setpoint temperature of 60°C. In the on/off control scheme, the heater power is maximum (100%) if the temperature is below 60°C and zero (0%) if the temperature is above 60°C. This could be achieved by using, for example, a mechanical thermostat.

Clearly, on/off control does not result in a stable temperature – it oscillates around the setpoint and does not settle down. This is caused by the time delay between power being applied to the heater and the temperature recorded by the thermocouple starting to increase. There is a similar time delay between power being removed from the heater and the temperature starting to fall. As a result, if the heater is turned on or off

when the temperature crosses the 60°C threshold, the temperature will overshoot or undershoot and oscillations occur.

Some means is needed of predicting the future temperature behaviour of the oven so, for example, the heater power can be decreased before the temperature rises above the 60°C setpoint. This can be achieved using PID control, the results of which are also shown in Fig. 6.

With correctly-tuned PID control, the temperature increases smoothly to 60°C and remains steady. Note how the heater power varies over time. Initially, a maximum of 80% power is applied to heat the metal block quickly. As the temperature begins to approach 60°C, the heater power is decreased until it remains steady at around 17%. If there are any temperature fluctuations, the PID algorithm automatically adjusts the heater power to try and restore a steady temperature.

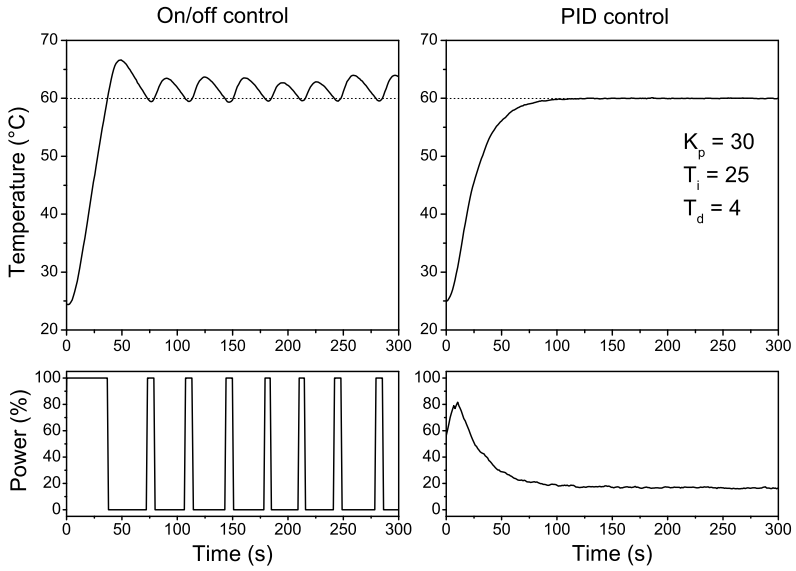



**Fig. 6: Comparison of on/off control (left-hand graphs) and PID control (right-hand graphs). Setpoint stepped from 25°C to 60°C, indicated by the dashed line. PID parameters: $K_p$=30, $T_i$=25, $T_d$=4.**

## 7.1 PID EQUATION

Table 1 lists definitions of the various terms used in the PID equation.

**Table 1: PID term definitions**

| Term | Description |
|---|---|
| *SV* | Setpoint value. The *required* temperature. |
| *PV* | Process value. The *actual* current temperature of the process. |
| *e* | Error. Difference between the setpoint and process values: $e = SV - PV$. When $SV > PV$, the error is positive. |
| *P* | Output power. This is the power applied to the heating element. *P* lies within the range 0% (fully off) to 100% (fully on). |
| $K_p$ | Proportional gain. Units: (% power) per (degree C). |
| $T_i$ | Integral time. Units: seconds |
| $T_d$ | Derivative time. Units: seconds |

$K_p$, $T_i$, $T_d$ are the three PID parameters which determine how the PID control loop responds to changes in the setpoint and process temperatures.

The PID control equation, shown below, is used to calculate the power which should be applied to the heating element. It consists of three terms, corresponding to the proportional, integral, and derivative actions.

$$P = K_p e + \frac{K_p}{T_i}\int_0^t e\,dt - K_p T_d \frac{d(PV)}{dt}$$

If the power, *P*, calculated from this equation is outside the valid limits of 0% to 100%, it is "clipped" to whichever limit is relevant.

Note that, although the above form of the PID equation is the most common, other sources sometimes use a different form, or alternative symbols for the various terms. However, the notation used here is self-consistent and matches that used in the accompanying software.

The contribution of each of the three terms is explained and demonstrated below.

### 7.1.1 PROPORTIONAL (P) CONTROL

If only proportional action is used, the control equation becomes:

$$P = K_p e$$

The applied power depends only on the error between the setpoint and process temperatures. Fig. 7 shows an example of proportional control with different values of $K_p$. As $K_p$ is increased, the steady-state temperature reached moves closer to the setpoint, but the temperature also becomes more unstable. Although, for example, $K_p$=3 results in a smooth increase to a stable temperature of around 55°C, $K_p$=10 results in an initial overshoot before the temperature settles to around 57°C. In fact, if $K_p$ is too large, oscillations can become permanent and the temperature never reaches a steady value. Very large values of $K_p$ (for example, greater than 1000) simulate on/off control, since the power changes rapidly from 0% to 100% as the temperature crosses the setpoint.

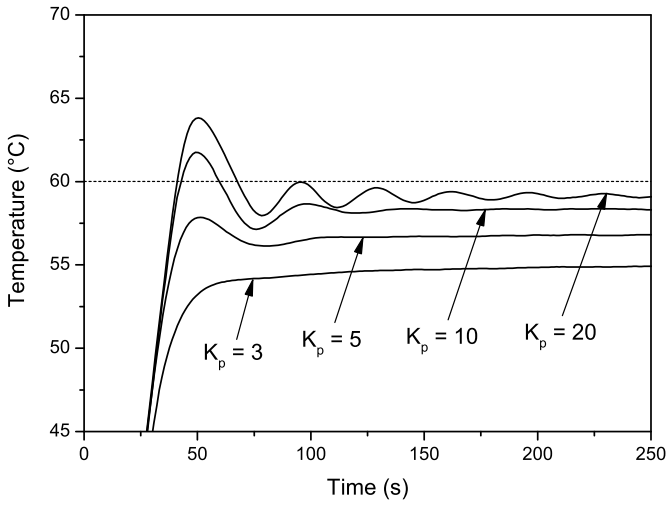



**Fig. 7: Proportional-only control for four different values of $K_p$.**
**Setpoint stepped from 20°C to 60°C, indicated by the dashed line.**

With proportional-only control, the steady-state temperature reached is always a certain amount (called the *droop*) below the setpoint temperature, which is clearly undesirable. This can be solved by the addition of the integral term.

### 7.1.2 PROPORTIONAL+INTEGRAL (PI) CONTROL

With proportional and integral action, the equation becomes:

$$P = K_p e + \frac{K_p}{T_i}\int_0^t e\,dt$$

By integrating the error over time, the integral term gradually compensates for the droop caused by the proportional term. It attempts to eliminate the droop within a time roughly equal to $T_i$. Smaller values of $T_i$ will result in more severe (faster) integral action. Fig. 8 shows an example of proportional+integral control for three different values of $T_i$.

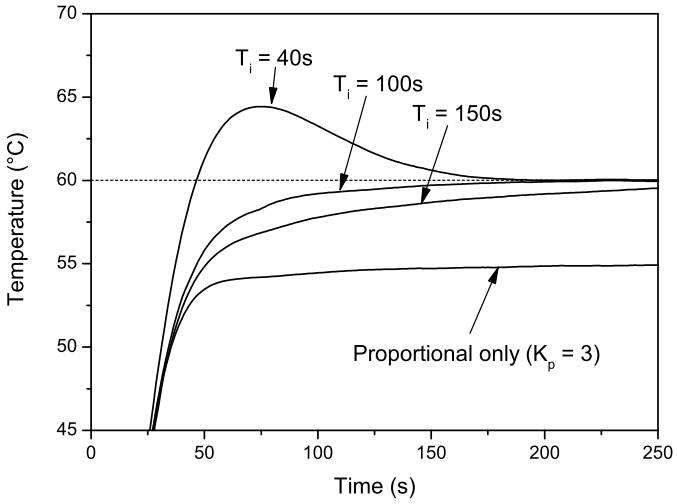



**Fig. 8: Proportional+integral control.**
**Setpoint stepped from 20°C to 60°C, indicated by the dashed line.**

When $T_i$ = 100s, droop is completely eliminated by around 200s. This would be an acceptable value of $T_i$ to use for a PI controller. A larger value, $T_i$ = 150s, results in an excessively long response time. With a smaller value, for example $T_i$ = 40s, the controller attempts to remove the droop too quickly and the temperature overshoots the setpoint before settling down. Just as too large a value of $K_p$ can cause oscillations, so can too small a value of $T_i$.

For most temperature control systems, proportional+integral control is sufficient, since it results (theoretically) in a temperature equal to the setpoint temperature. However, a faster response can sometimes be achieved by adding the derivative term.

### 7.1.3 PROPORTIONAL+INTEGRAL+DERIVATIVE (PID) CONTROL

With all three terms included, the equation becomes:

$$P = K_p e + \frac{K_p}{T_i}\int_0^t e\,dt - K_p T_d \frac{d(PV)}{dt}$$

The addition of the derivative term slows the rate of change of heater power. For example, if the temperature is below the setpoint and increasing rapidly, the derivative term is negative and results in a decrease in the applied power, which in turn helps to slow down the temperature rise. The derivative term can be used to correct overshoot or oscillations caused by the integral term, as shown in Fig. 9. Because this allows the use of a smaller value of $T_i$, addition of the integral term can result in a faster system response. However, because the derivative term is based on the process temperature, it can be sensitive to noise and should not be used in noisy environments.

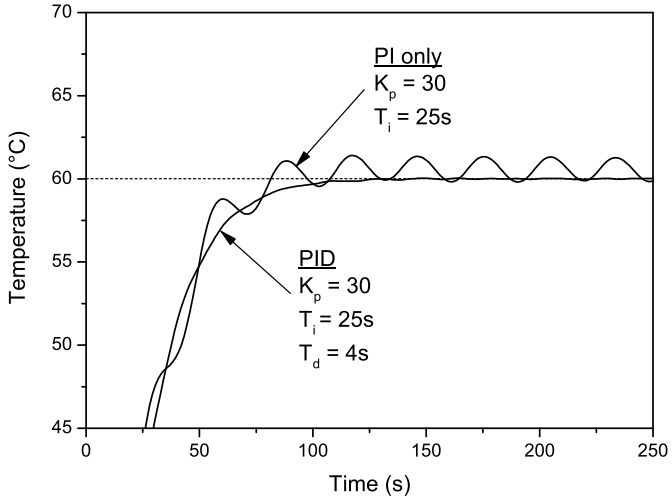



**Fig. 9: Proportional+integral+derivative control. Addition of the derivative term removes the oscillations. Setpoint stepped from 20°C to 60°C, indicated by the dashed line.**

## 7.2 TUNING

As can be seen from the above examples, PID control only works when the parameters $K_p$, $T_i$, $T_d$ are chosen correctly. This process is called *tuning* and can be performed manually. Tuning is best performed at or near the temperature of interest, since the parameters can themselves depend on temperature.

It was mentioned above that, if $K_p$ is too large, oscillations can occur. The smallest value of $K_p$ which causes sustained temperature oscillations in a proportional-only control system is called the *critical gain*, $K_c$, and the period of the oscillations is $T_c$. Once $K_c$ and $T_c$ have been determined, using either of the methods below, suitable values for the parameters $K_p$, $T_i$, $T_d$ can be calculated from them.

### 7.2.1 CLOSED-LOOP

The controller is placed in proportional only mode and a setpoint temperature chosen – in the example given below, this is 60°C.

A trial value for $K_p$ is then chosen. It is nearly impossible to tell what a suitable value is beforehand and it is best determined by trial and error.

The controller is switched on and the temperature profile observed for a sufficient length of time to determine whether sustained oscillations are occurring. This can sometimes be difficult to determine, but 10-20 oscillation cycles are usually enough.

If oscillations do occur, $K_p$ should be decreased. Conversely, if no oscillations (or damped oscillations) are observed, $K_p$ should be increased. The controller should be switched off and the temperature allowed to fall sufficiently before performing the next test. The aim is to determine the smallest value of $K_p$ which produces *sustained* oscillations. The exact value is not too critical. When sustained oscillations are observed, the peaks of the wave tend to "top-out" at the setpoint temperature.

Fig. 10 shows an example of the tuning process for three values of $K_p$, demonstrating damped oscillations ($K_p$ = 20), sustained oscillations ($K_p$ = 27), and excessive oscillations ($K_p$ = 50). The value of the critical gain is $K_c$ = 27 % deg$^{-1}$ and the corresponding oscillation period is $T_c$ = 29s.

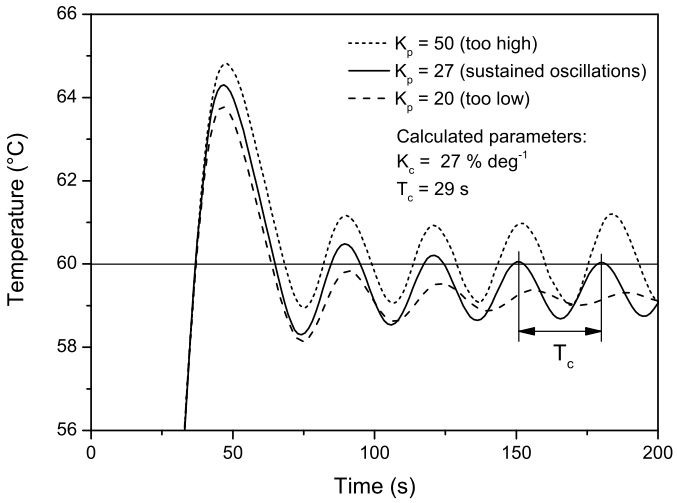



**Fig. 10: Example of closed-loop tuning. Setpoint stepped from 20°C to 60°C each time.**

### 7.2.2 OPEN-LOOP

An alternative tuning method is based on measuring the temperature profile as the heater power is stepped between two values.

The process temperature should first be brought up to a value near the desired tuning point and allowed to stabilise. This can be done either by applying a fixed power and waiting until a steady-state temperature is reached, or by using extremely conservative PI-only control, which can be much faster. For the IJ-7 demo module, suitable PI parameters are $K_p$ = 3, $T_i$ = 40. Note that PID control is less useful, because of the noise introduced into the recorded power reading by the derivative term, making it harder to determine the initial power level.

Once the temperature is stable, the power is increased by a certain amount. The actual amount is not critical, but should be enough that a clear, noise-free temperature profile is obtained. If desired, several different increases can be tried and the results averaged. Once the temperature has risen past the curve's point of inflection (maximum rate of rise), the controller can be switched off. If PI control was used to stabilise the temperature initially, power can be increased by switching to manual power control and entering a suitable value.

Fig. 11 shows an example of the process. The temperature is initially held at 60°C using PI-only control ($K_p$ = 3, $T_i$ = 40). From a graph of the power, this corresponds to an initial heater power of 15.8%. At time t = 0s, the heater power is stepped to 60% and the temperature starts to rise. At t = 24s, the temperature is rising at a

maximum rate, *R*, of 0.506 °C $s^{-1}$. A tangent line is drawn at the point of maximum rate and the time of its intersection with the original steady-state temperature measured. In Fig. 11, this *lag time*, *L*, is 6.6s.

(Instead of drawing a tangent line and measuring the lag time graphically, it can be calculated from

$$L = t_{max} - \left(\frac{T_{max} - T_{init}}{R}\right)$$

where $t_{max}$ is the time at the inflection point, $T_{max}$ is the temperature at the inflection point, $T_{init}$ is the initial steady-state temperature, and R is the maximum rate of rise, measured at the inflection point.)

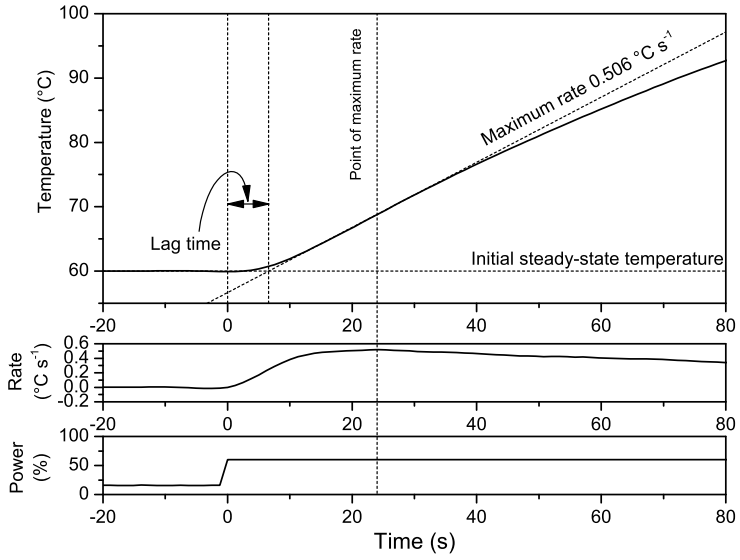



**Fig. 11: Example of open-loop tuning, showing temperature, rate, and power versus time. Rate is the derivative of the temperature curve.**

The critical gain and associated oscillation period can be calculated from the maximum rise rate and lag time using:

$$K_c = \frac{2\Delta P}{RL} \quad \text{and} \quad T_c = 4L$$

*ΔP* is the *absolute* change in heater power during the step. In this example, the power was increased from 15.8% to 60%, so *ΔP* = 44.2%. Using the values of *R* = 0.506 °C $s^{-1}$ and *L* = 6.6s, we can calculate $K_c$ = 26.5 % $deg^{-1}$ and $T_c$ = 26.4s. These values are very close to those obtained from the closed-loop tuning method.

As shown in Fig. 12, the critical gain and period do not depend appreciably on the final power chosen in the tuning method. Stepping to a value in the region of 60%-100% is adequate.

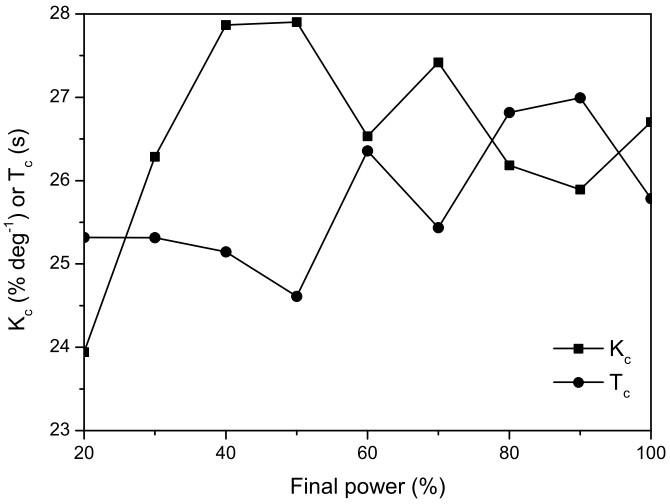



**Fig. 12: Critical gain and period versus final power used in open-loop tuning**

Open-loop tuning is sometimes quicker, since only one measurement is needed, but does rely on achieving a stable initial temperature. Closed-loop tuning is more reliable, but does require some trial and error to determine an optimum value for the critical gain. Choice of which method to use is left entirely to the user.

### 7.2.3 SUMMARY + TABLE OF TUNING FACTORS

Once the critical gain and period have been determined, using either tuning method, suitable values for $K_p$, $T_i$, $T_d$ can be calculated using the multiplying factors listed in Table 2. Various control schemes are listed, differing in the degree of control they provide. Ziegler-Nichols is very common and provides quite a rapid response. However, this can cause some overshoot, especially with fast changes in setpoint. Tyreus-Luyben provides a softer, more stable response for systems which have fixed or slowly-varying setpoints.

**Table 2: PID tuning factors**

| Control scheme | | $K_p/K_c$ | $T_i/T_c$ | $T_d/T_c$ |
|---|---|---|---|---|
| Ziegler-Nichols | P | 0.5 | - | - |
| | PD | 0.6 | - | 0.125 |
| | PI | 0.45 | 0.85 | - |
| | PID | 0.6 | 0.5 | 0.125 |
| Tyreus-Luyben | PI | 0.31 | 2.2 | - |
| | PID | 0.45 | 2.2 | 0.16 |
| Shinskey | P | 0.5 | - | - |
| | PI | 0.5 | 0.45 | - |
| | PID | 0.25 | 0.5 | 0.12 |
| General | Pessen Integral Rule | 0.7 | 0.4 | 0.15 |
| | Some Overshoot | 0.33 | 0.5 | 0.33 |
| | No Overshoot | 0.2 | 0.5 | 0.33 |

A major advantage of calculating $K_p$, $T_i$, $T_d$ using the factors from Table 2 is the possibility of varying the parameters depending on the current process temperature, a process called *gain scheduling*. If a list of values of the critical gain and period at different temperatures is known, $K_p$, $T_i$, $T_d$ can be calculated at any temperature by using the factors. Although it can be time-consuming to measure the critical gain and period over a range of different temperatures, it can sometimes result in greatly improved control, especially when the system is operated over a wide temperature range.

# 8 HARDWARE REFERENCE

## 8.1 REAR PANEL

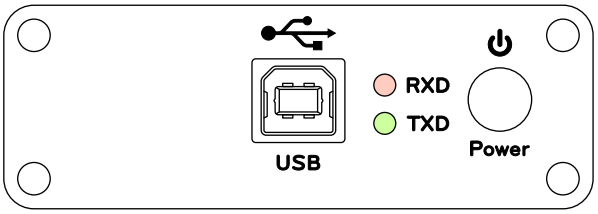



**Fig. 13: Rear panel view**

### 8.1.1 USB CONNECTION

Connect either directly to a computer or to a powered USB hub. Do not attempt to connect to an unpowered hub, as the hub will be unable to supply sufficient power to the controller.

### 8.1.2 TXD/RXD INDICATORS

These two LEDs indicate the flow of data between the computer and the device as follows:

RXD (red) – data *received* by the device *from* the host computer

TXD (green) – data *transmitted* by the device *to* the host computer

### 8.1.3 POWER SWITCH

Turns the device on or off. Illuminated green when powered.

## 8.2 FRONT PANEL

See Fig. 14 for a view of the instrument's front panel. It consists of an 8-way pluggable terminal block for connections to the external system, three status indicators, and a universal thermocouple connector.

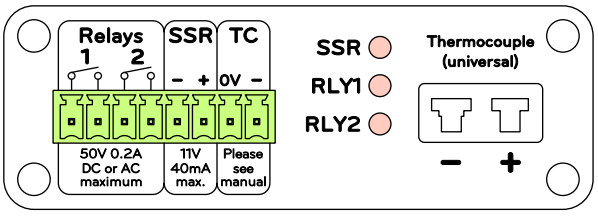



**Fig. 14: Front panel view**

### 8.2.1 GENERAL-PURPOSE RELAY OUTPUTS

Two general-purpose, normally-open relay outputs are provided, capable of switching a maximum voltage of 50V DC or AC at a maximum current of 0.2A. The on resistance is typically in the range 2Ω - 10Ω. The relay outputs are obtained from an ASSR-1228 dual solid-state relay (see Fig. 15) and are therefore optically isolated from the rest of the circuitry. For safety, it is recommended that the maximum voltage at the relay outputs relative to ground be kept to below 50V.

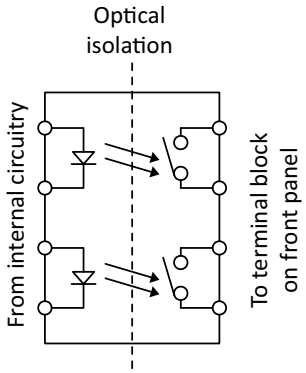



**Fig. 15: ASSR-1228 solid-state relay (internal)**

The relay outputs can be used, for example, to drive an alarm as shown in Fig. 16.

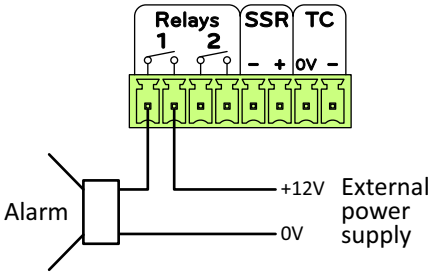



**Fig. 16: Using a relay output to drive an alarm**

It is possible to replace the internal solid-state relay, should it become damaged – please contact support if this is required.

### 8.2.2 MAIN SOLID-STATE RELAY OUTPUT

For safe operation, you **must always** use a relay, optoisolator, or other isolation element between the main SSR output of the temperature controller and the external system, and **not** connect the main SSR output of the temperature controller directly (for example, to a MOSFET or transistor switch). The **only exception** is the IJ-7 demonstration module, which can be connected directly.

This provides a variable duty cycle output suitable for driving a relay or optoisolator. The output voltage is around 11V open-circuit, dropping to 10V when delivering the maximum output current of 40mA, as shown in Fig. 17. Output resistance is around 30Ω.

The cycle time can be adjusted from approximately 0.5s to 60s via the software. Longer cycle times should be used for systems which respond slowly to changes in heater power. If a mechanical relay (instead of a solid-state relay) is used instead, long cycle times help to prolong the contact life since the switching frequency is reduced.

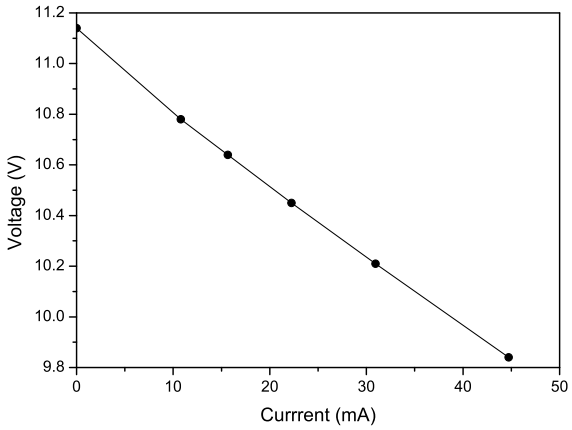



**Fig. 17: SSR output voltage and current**

Fig. 18 shows an example of connecting the temperature controller to a "brick-type" solid-state relay. A reed relay or optoisolator can also be connected instead, providing the current draw is kept below the maximum of 40mA.

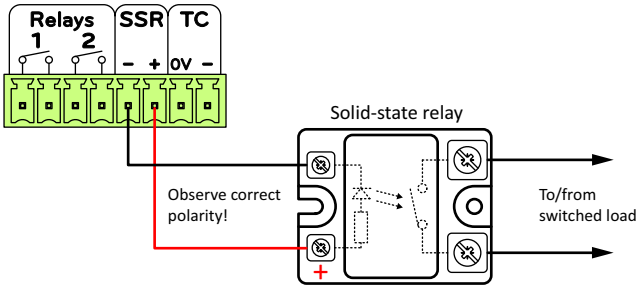



**Fig. 18: Connections for external solid-state relay**

### 8.2.3 THERMOCOUPLE GROUND REFERENCE

A suitable thermocouple ground reference must be chosen, depending on whether the thermocouple is floating or grounded. The two rightmost terminals of the terminal block, marked "0V" and "-", are used for this purpose. "0V" is connected to the negative supply rail of the temperature controller and hence to the negative supply of the USB port. "-" is connected directly to the negative terminal of the thermocouple connector. Depending on how these terminals are connected, either a floating or grounded thermocouple can be used, as described below.

If the thermocouple is

- bare (e.g. welded-bead) and not attached to anything (e.g. placed in mid-air to measure air temperature)
- bare and attached to an object, but the object itself is floating (not grounded)

then the thermocouple is "floating" relative to ground and the connection shown in Fig. 19 should be used. A shorting link is placed between "0V" and "-", thus connecting the negative terminal of the thermocouple to 0V.

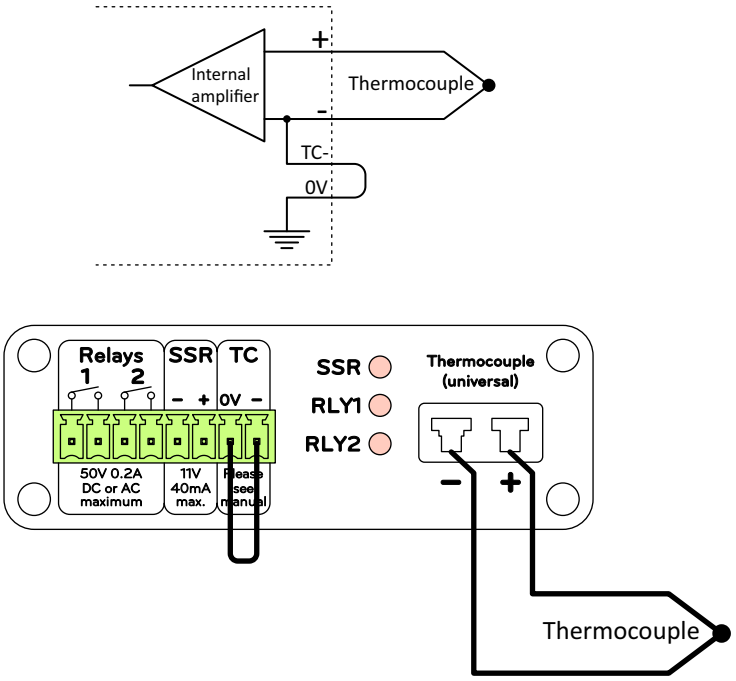



**Fig. 19: Floating thermocouple connections**

If, however, the thermocouple *is* connected to an object which is grounded, then the connection shown in Fig. 20 should be used. Here, a connection is made between "0V" on the controller and a location near where the thermocouple junction is attached to the remote object under measurement. This equalizes the grounds of both the temperature controller and the object.

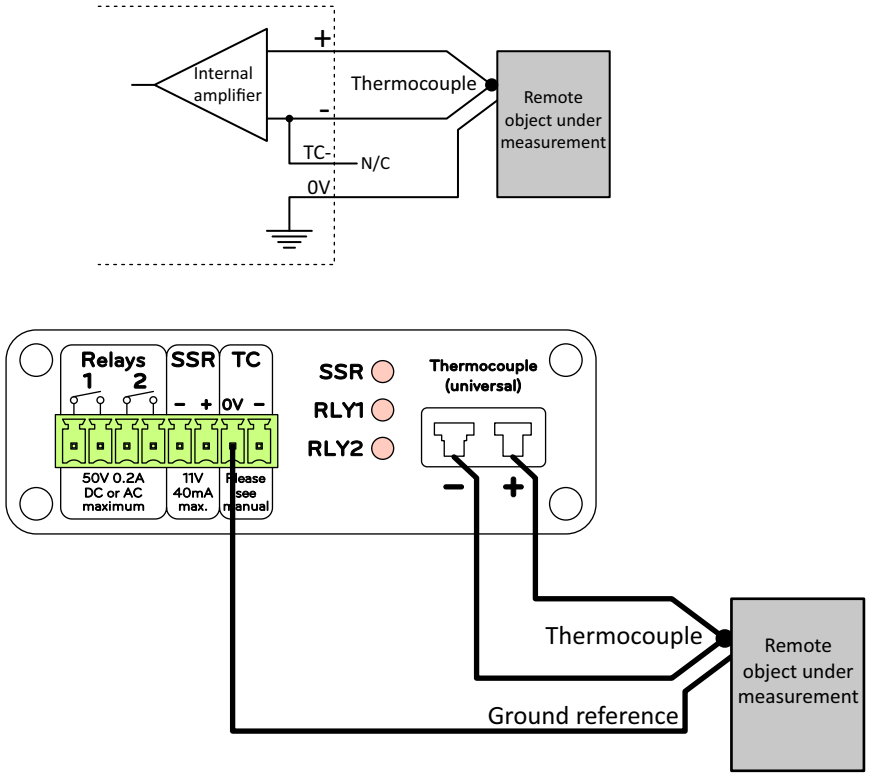



**Fig. 20: Remote grounded thermocouple connections**

Floating thermocouples result in better performance since no noise or ground-leakage currents can flow in the sensor leads and cause an error in the voltage reading.

### 8.2.4 STATUS INDICATORS

These indicate the status of the main solid-state relay output (ON = SSR output high) and both of the general-purpose relay outputs (ON = contacts closed).

### 8.2.5 THERMOCOUPLE INPUT

The unit can accept any type of thermocouple, since the connector uses plain copper contacts and cold-junction compensation is performed. The maximum differential input is -30mV to +80mV. Maximum absolute voltage on either terminal is +/- 3V relative to ground.

Note that the choice of thermocouple *is* restricted by the types supported in the software (currently E, J, K, N, or T).

The plugs on some thermocouples are sometimes very tight to insert - take care not to stress the plug during insertion. The plug does not have to be inserted fully – just ensure that the plug terminals make contact with the socket.

## 9 SOFTWARE REFERENCE

Subsequent sections outline the use of each region of the program's interface.

### 9.1 POWER CONTROL

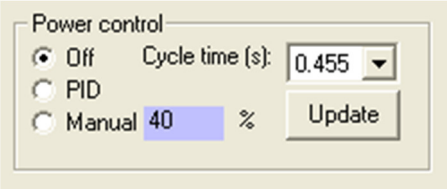



**Fig. 21: Power and cycle time control**

The three option buttons at the left select the power control mode. If "Off", the output power is set to zero and the heater permanently held off. If "PID" is selected, the program performs PID calculations and sets the output power accordingly. If "Manual" is selected, the output power is set equal to the value in the blue box. Power can range from 0% (fully off) to 100% (fully on).

If it is only desired to record or view temperature (no power control), then the power control mode can be set to "Off" and all the other functions of the program (file logging, graphical display) will still work as normal.

"Cycle time". This controls the cycle time used by the temperature controller for switching the solid-state relay output. It can be adjusted from approximately 0.5s to 60s. For example, if the cycle time is 1.82s and the power output level is 20%, the relay output will be high (on) for 1.82*0.2 = 0.36s and low (off) for 1.82*0.8=1.46s. The percentage on time remains constant, regardless of what cycle time is selected. Short cycle times are used for systems with a fast thermal response and are best suited to solid-state relays because of the high switching frequency. Conversely, longer cycle time are used for slower systems and mechanical relays where contact wear is less of a problem.

To change the cycle time, choose the desired time from the drop-down list and click "Update" (Fig. 22). The software transmits the new cycle time to the controller.

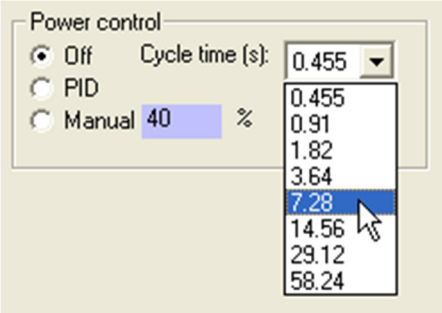



**Fig. 22: Setting cycle time**

## 9.2 ADC SETTINGS

See Fig. 23. These control the settings of the analogue-to-digital converter (ADC) inside the controller. They should not need adjustment, except when first running the software or when performing calibration.

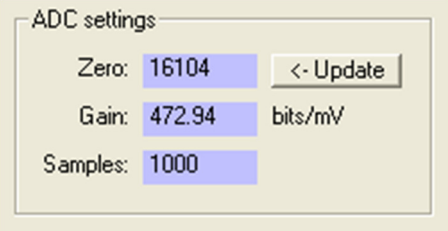



**Fig. 23: ADC settings**

"Zero" is the reading returned by the ADC when zero voltage is present at the thermocouple connector input. A value can be entered manually, or the current ADC value can be loaded by clicking the "<- Update" button.

"Gain" is the sensitivity of the ADC to a thermocouple voltage and is measured in bits per mV.

"Samples" controls the degree of averaging performed. The controller takes this number of samples and averages them to reduce noise. More samples results in lower noise, but may take slightly longer to record. Unless used in extremely noisy environments, a value of 1000 samples is a good compromise. It takes about 150µs to record one sample, so 1000 samples takes 0.15s. The maximum number of samples is 65535.

When the program is first run, both Zero and Gain will be set to default values and must be changed to match the individual controller used. Refer to the calibration sheet provided with the controller and enter the Zero and Gain values specified. For further details on calibration, see Section 10.

## 9.3 CURRENT VALUE DISPLAY

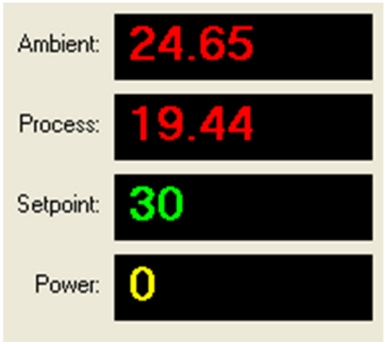



**Fig. 24: Current value display**

"Ambient" – the temperature of the actual controller, specifically, the temperature of the thermocouple connector. This is needed for performing cold-junction compensation in order to calculate the thermocouple temperature.

"Process" – current temperature of the process, as measured by the thermocouple. Measured in °C.

"Setpoint" – setpoint temperature. Measured in °C.

"Power" – current output power, range 0% to 100%.

## 9.4 THERMOCOUPLE TYPE

The software can accept various types of thermocouples. Some of the most common are available from the drop-down list, as shown in Fig. 25.

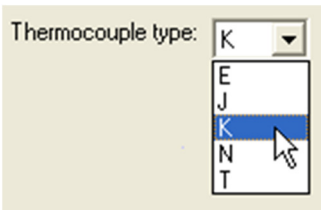



**Fig. 25: Set thermocouple type**

## 9.5 LOGGING

The software can record time, temperature and output power to a tab-delimited log file (Fig. 26).

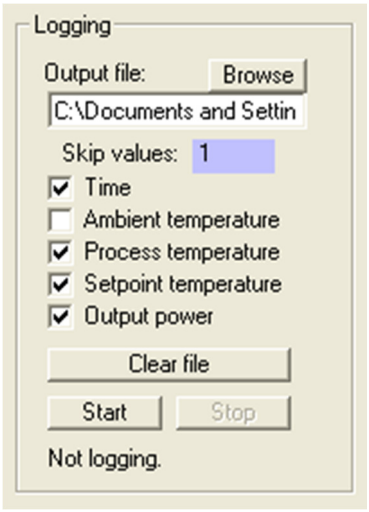



**Fig. 26: File logging**

The destination file can be selected by clicking "Browse". The filename entered should end with ".txt". Data is stored in a tab-separated format and can easily be imported into programs such as Microsoft Excel or Microcal Origin for processing. The user can choose which values to record to the file by ticking the appropriate boxes. An example of a log file, after being imported into Excel, is shown in Fig. 27. If a file already contains log information, the new log is appended to the end, as shown.

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | Logging started at 08/05/2012 21:05:06 | | | | |
| 2 | Time | Ambient | Process | Setpoint | Power |
| 3 | 0 | 20.09375 | 2572.347 | 30 | 0 |
| 4 | 0.972 | 20.0625 | 2572.174 | 30 | 0 |
| 5 | 1.94 | 20.0625 | 2572.065 | 30 | 0 |
| 6 | 2.873 | 20.09375 | 2572.13 | 30 | 0 |
| 7 | 3.818 | 20.0625 | 2572.065 | 30 | 0 |
| 8 | 4.749 | 20.09375 | 2572.238 | 30 | 0 |
| 9 | 5.767 | 20.0625 | 2572.174 | 30 | 0 |
| 10 | Logging stopped at 08/05/2012 21:05:13 | | | | |
| 11 | Logging started at 08/05/2012 21:05:17 | | | | |
| 12 | Time | Ambient | Process | Setpoint | Power |
| 13 | 0 | 20.0625 | 2572.174 | 30 | 0 |
| 14 | 1 | 20.0625 | 2572.282 | 30 | 0 |
| 15 | 1.937 | 20.09375 | 2572.347 | 30 | 0 |
| 16 | 2.944 | 20.0625 | 2572.282 | 30 | 0 |
| 17 | Logging stopped at 08/05/2012 21:05:21 | | | | |

**Fig. 27: Example of log file**

The time in the log file is the time elapsed (in seconds) since logging was started. Note that it often does not increase in uniform increments. This is because each cycle of the program may take a slightly different time to run, depending on the resource usage of the computer. If other programs are also running, or CPU usage is high, this may increase the interval between log entries slightly, but will not affect the temperature control (see Section 11.3).

By default, a log entry is recorded every time the program completes a cycle (roughly every second). For slowly-varying processes, this can lead to a large amount of recorded data. To prevent this, the log file can "skip" values, as specified in the "Skip values" box. For example, if "3" is entered, the log file only records once in every three program cycles.

"Clear file" allows the user to erase the current log file.

"Start" and "Stop" control the log file operation. The current status (Logging/Not Logging) is displayed beneath the buttons.

## 9.6 PID PARAMETERS AND GAIN SCHEDULING

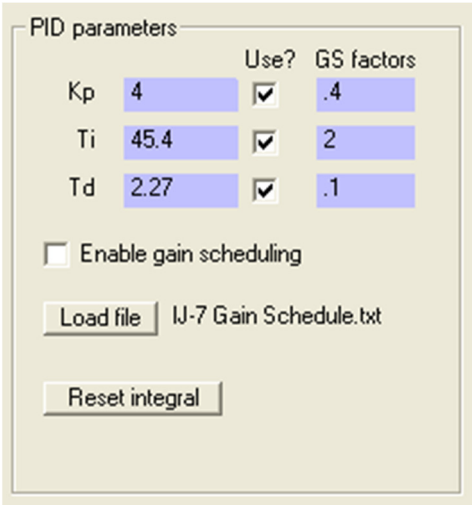



**Fig. 28: PID parameters and gain scheduling**

Here, the user can set values for the proportional gain ($K_p$), integral time ($T_i$) and derivative time ($T_d$). The check boxes under "Use?" determine which of the terms in the PID equation (proportional, integral, or derivative) the program uses. For example, if only the top check box is ticked, the program functions as a proportional-only controller and ignores the values of $T_i$ and $T_d$.

It is possible to perform gain scheduling (see Section 7.2.3), where the PID parameters are changed depending on the current process temperature. To enable gain scheduling, first create a text file (using any text editor, such as Notepad) which contains the gain scheduling information. An example of such a file is shown in Fig. 29. Each line should contain the process temperature (in °C) and the critical gain & oscillation period at that temperature, separated with commas. For example, in Fig. 29, $K_c$ = 10 % deg$^{-1}$ and $T_c$ = 22.7s at a temperature of 60°C. The temperatures must be presented in increasing order.

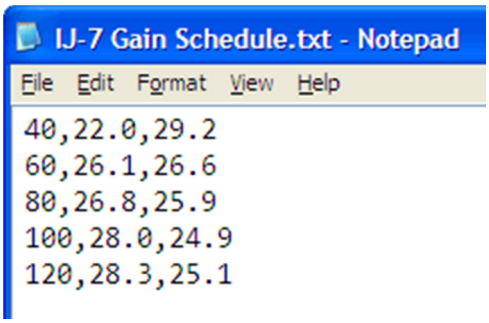



**Fig. 29: Example of a gain scheduling file**

The temperature range covered by the entries in the file should cover the expected operating temperature range of the controlled system. Based on the file entries, the software performs linear interpolation to determine $K_c$ and $T_c$ at the current process temperature, then uses a set of factors to calculate suitable values for the PID parameters $K_p$, $T_i$ and $T_d$.

Click “Load file” and select the gain scheduling file to use. Gain scheduling can be enabled/disabled using the checkbox.

The factors used to calculate $K_p$, $T_i$ and $T_d$ can be entered under “GS factors”. See Table 2, page 17, for a list of factors.

The “Reset Integral” button can be used to manually re-zero the PID integral term. This can sometimes be useful if $T_i$ has been changed by a large amount, or if the manual setpoint has suddenly changed, both of which will cause a large fluctuation in the integral term.

## 9.7 SETPOINT AND PROGRAM CONTROL

**Fig. 30: Setpoint & program control (shown with a program running)**

The setpoint temperature can either be set manually, or can be varied according to a user-defined program. If there is no program running, then the software automatically uses the setpoint value in the “Manual setpoint” box.

A sample temperature profile is shown in Fig. 31 and the associated program in Fig. 32.

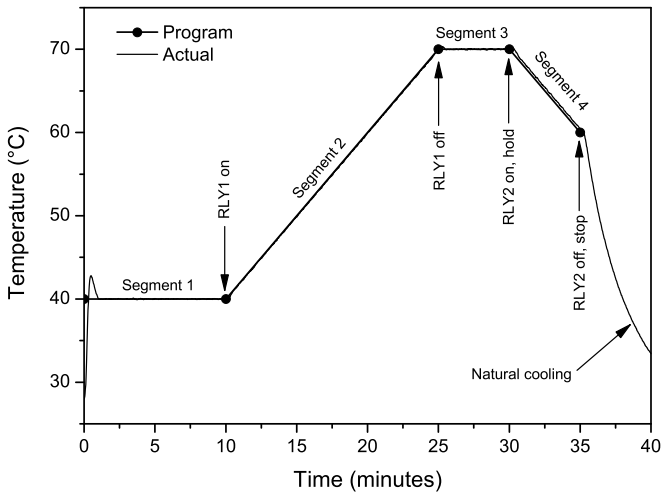



**Fig. 31: Example temperature program, showing both the programmed setpoint value and the actual temperature reached.**

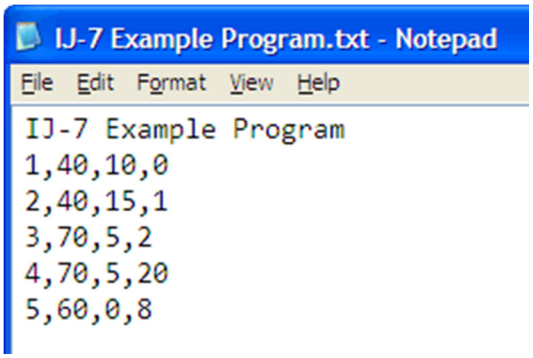



**Fig. 32: Program to produce profile shown in Fig. 31**

The first line can contain any text for the user's reference – title, date etc. It must *not* be blank, however.

Each segment of the temperature profile is defined by a line in the file. Each line contains four numbers, separated by commas, as defined below:

Segment number, start temperature, segment duration, control code

**Segment number**

Indicates which segment the line refers to.

**Start temperature**

The temperature (in °C) at which the segment starts. The temperature is ramped up or down from the start temperature of the *current* segment to the start temperature of the *next* segment. If the start temperatures of the current and next segments are identical, then the temperature is held constant.

**Segment duration**

The time (in minutes) for which the segment lasts. To stop the program at the beginning of the current segment, enter zero duration.

**Control code**

This is used to control the relay outputs and/or to pause the program. The control code is calculated by adding up the numbers corresponding to each desired operation, listed in Table 3.

**Table 3: Program control codes**

| Code | Operation |
|---|---|
| 1 | Turn relay 1 on |
| 2 | Turn relay 1 off |
| 4 | Turn relay 2 on |
| 8 | Turn relay 2 off |
| 16 | Pause program |

For example, if the user wants to turn relay 2 on and pause the program, the control code is 4 + 16 = 20. The operations (if any) defined by the control code are performed at the start of the current segment. If no operations are desired, the control code is 0.

Looking at the program, we can analyse its behaviour:

Segment 1: Temperature is held constant at 40°C for 10 minutes.

Segment 2: The control code is 1, so relay 1 is turned on. Temperature rises to 70°C over 15 minutes.

Segment 3: The control code is 2, so relay 1 is turned off. Temperature remains constant at 70°C for 5 minutes.

Segment 4: The control code is 20 (= 4 + 16), so relay 2 is first turned on, then the program is paused and waits for the user to resume. After resuming, the temperature falls to 60°C over 5 minutes.

Segment 5: The control code is 8, so relay 2 is turned off. Since the segment duration is zero, the program stops.

To use a program, load it by clicking the button and then click "Start". When running, the current segment is displayed. The program can be stopped or paused at any time by clicking "Stop" or "Hold". To resume a paused program, click "Start" again.

## 9.8 RELAY CONTROL

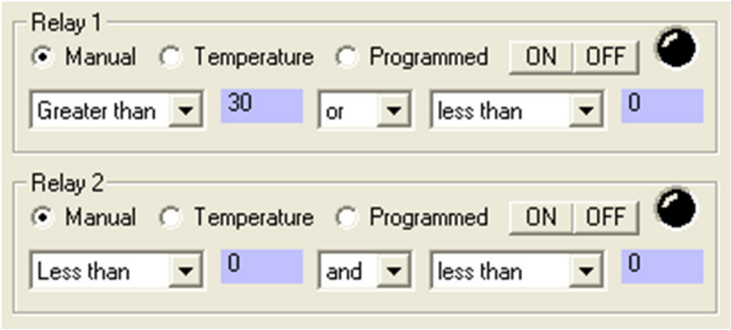



**Fig. 33: Relay control**

The user can control both of the general-purpose relay outputs (for hardware details, see Section 8.2.1). Three operation modes are available: manual, temperature, or programmed.

**Manual**

In manual mode, the relays can be controlled by clicking the appropriate on/off buttons for each.

**Temperature**

The relays are controlled depending on the current process temperature. For example, if we wanted relay 1 to be on when the temperature is between 50°C and 100°C, the drop-down lists and boxes should be configured to read:

Greater than 50 and less than 100

Alternatively, if we wanted the relay *off* between 50°C and 100°C (and on everywhere else), we would choose:

Less than 50 or greater than 100

**Programmed**

The relay outputs are placed under program control and can be switched using appropriate values of the control code (see Section 9.7).

## 9.9 GRAPHICAL DISPLAY

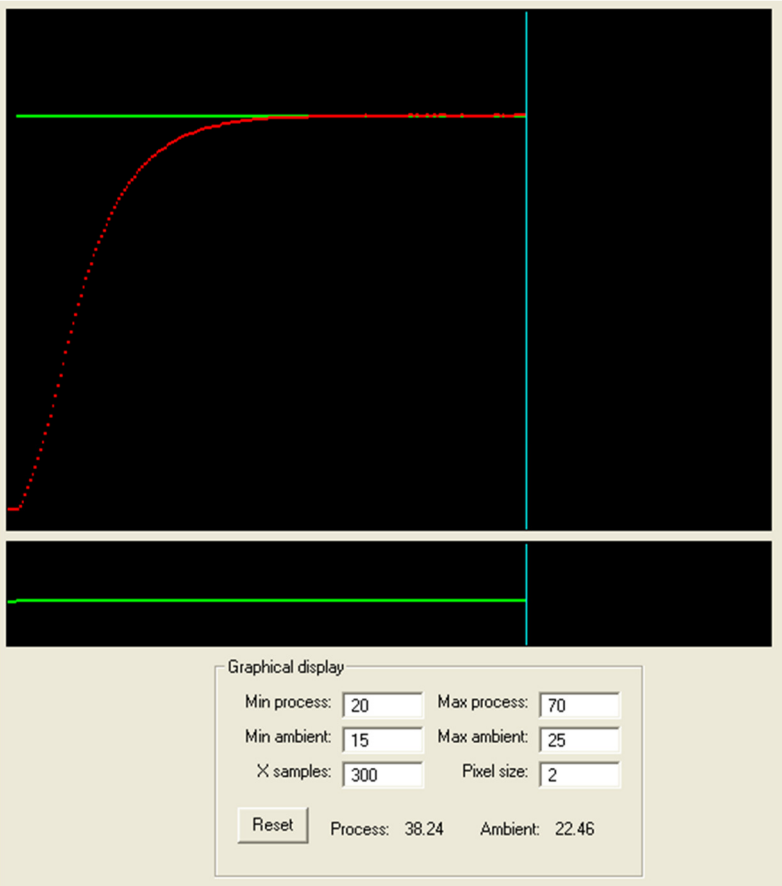



**Fig. 34: Graphical display**

The upper (larger) display shows both the setpoint temperature (green) and process temperature (red). The lower display shows the ambient temperature. Once the displayed data reaches the right-hand edge of the display, it starts again from the left, overwriting the old data. The blue vertical line indicates the current data point.

If the mouse hovers over either of the display areas, the temperature at the mouse location (either process or ambient, depending on which display the mouse is over) is shown next to “Process:” or “Ambient:” in the panel below the displays.

The display ranges can be set by the values in the boxes below the display. “Min process:”/“Max process:” are the minimum and maximum process temperatures shown on the upper display. “Min ambient:”/“Max ambient:” are the minimum and maximum ambient temperatures shown on the lower display. “X samples:” determines how many data points are shown on the display. For slow processes, the number of samples can be increased to fit on the display. “Pixel size:” adjusts the size of the displayed data points.

To update the display parameters, click “Reset”. The display will clear and start recording with the new parameters.

# 10 CALIBRATION

Calibration consists of determining two factors: the ADC zero value and the ADC gain. When performing calibration, the thermocouple ground reference pins should be connected as shown in Fig. 19 for a floating thermocouple.

## 10.1 ADC ZERO

ADC zero calibration can be performed without the need for any additional equipment. Using a short piece of solid-core wire (approximately 0.6mm diameter), short both pins on the thermocouple connector together to ensure there is zero voltage present.

In the software, enter a reasonably large value (for example, 50000) into the “Samples:” box under the ADC settings section (Section 9.2). Click “<- Update” and wait for the value in the “Zero:” box to clear and reappear. This is the ADC zero value.

Set the “Samples:” back to 1000 or whatever value was being used previously.

## 10.2 ADC GAIN

In order to determine the ADC gain, a precision voltage reference must be available. The exact voltage is not critical, but it should be in the range 30-50mV to obtain an accurate calibration. It should not exceed 70mV.

Connect the precision voltage reference to the thermocouple connector, making sure of the polarity. Again, in the software, enter a large value for the “Samples:” and click the “<- Update” button. To calculate the ADC gain, subtract the ADC zero from the value shown in the “Zero:” box (with the voltage reference applied) and divide by the voltage reference (in mV).

For example, suppose the ADC zero value (with the thermocouple connector shorted) is 16075. When a 50mV reference is connected, the ADC reading is 39836. The ADC gain is therefore (39836-16075)/50 = 475.22 bits/mV.

# 11 MISCELLANEOUS

## 11.1 DISCONNECTING THE CONTROLLER

If the controller is accidentally switched off or disconnected, the software may display an error message similar to that shown in Fig. 35 and will then quit. Ensure the controller is reconnected properly and switched on, then restart the software.

**Fig. 35: Error message shown on disconnect**

## 11.2 MAXIMUM RECORDING TIME LIMIT

The internal variable used to determine the time of the entries in the log file has a maximum value of approximately 25 days and then rolls over to zero. This may cause problems if data is recorded continuously for greater than 25 days.

## 11.3 HEAVY CPU USAGE

Although the software makes use of timing functions and does run slower if CPU usage is high, this should not affect the degree of temperature control in any way. Fig. 36 shows an example of this - high CPU usage causes large fluctuations in the calculation cycle time, but this does not affect the temperature.

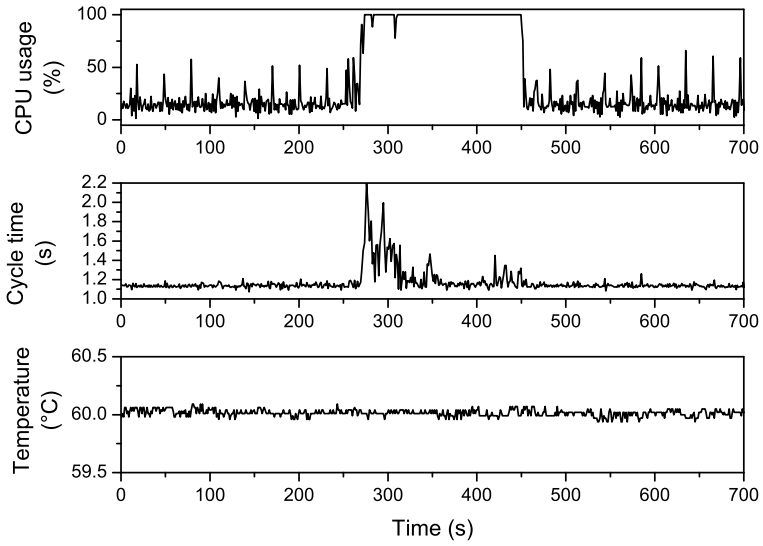



**Fig. 36: Immunity to high CPU usage (watching four YouTube videos simultaneously, listening to streaming radio, and converting an HD video file)**

## 11.4 OPERATING TEMPERATURE VARIATIONS

The controller is supplied calibrated at an ambient temperature of 20°C. Operation at other temperatures will result in a very small error, which consists of both an offset and a magnitude error.

The offset error is shown in Fig. 37. At lower temperatures, the thermocouple voltage as measured by the controller will *over-estimate* the true voltage, and vice versa at higher temperatures. The corresponding error in temperature depends on which type of thermocouple is connected. As an example, the +60µV error shown at -10°C in Fig. 37 would correspond to a temperature error of about +1.5°C for a K-type thermocouple (sensitivity = 40µV/°C).

The magnitude error is shown in Fig. 38 as a percentage of the actual thermocouple voltage. For example, at -10°C, the error is about +0.1%. This means that, if the actual thermocouple temperature was 100°C, the measured temperature would be 100.1°C (ignoring the offset error).

If prolonged operation at a temperature significantly different from 20°C is intended, the controlled should be recalibrated at the intended operating temperature for maximum accuracy.

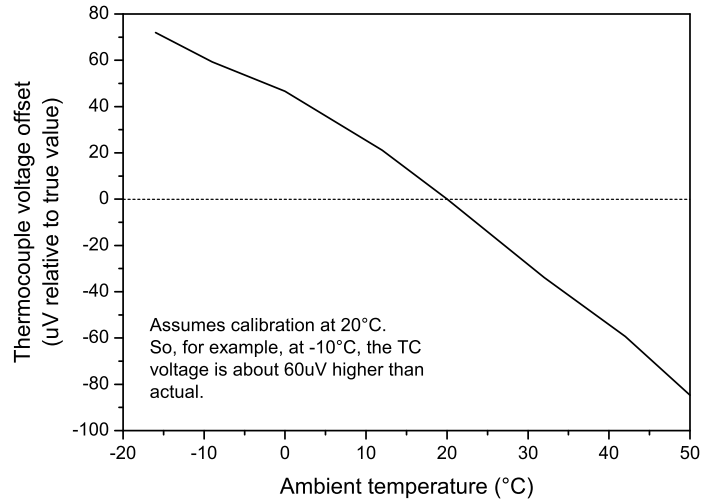

Thermocouple voltage offset
(uV relative to true value)
Ambient temperature (°C)
Assumes calibration at 20°C.
So, for example, at -10°C, the TC
voltage is about 60uV higher than
actual.

**Fig. 37: Thermocouple offset error vs. ambient temperature**

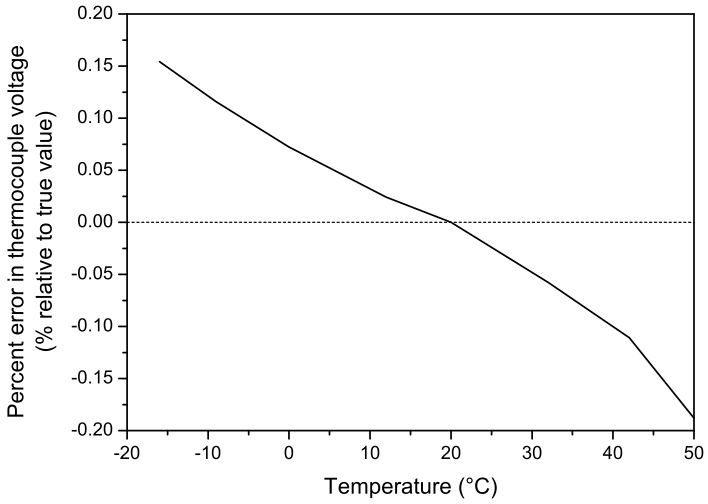
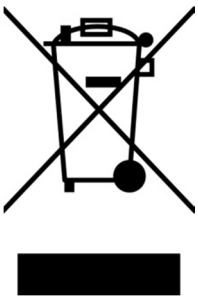



**Fig. 38: Thermocouple magnitude error vs. ambient temperature**

## 12 PRODUCT DISPOSAL INSTRUCTIONS



The symbol shown here and on the product means that the product is classified as Electrical or Electronic Equipment and should not be disposed with other household or commercial waste at the end of its working life.

The Waste of Electrical and Electronic Equipment (WEEE) Directive (2002/96/EC) has been put in place to recycle products using best available recovery and recycling techniques to minimize the impact on the environment, treat any hazardous substances and avoid the increasing landfill.

When you have no further use for it, please dispose of the product as per your local authority's recycling processes. Business users should ensure that this product is not mixed with other commercial waste for disposal.

## 13 CONTACT/SUPPORT

Dr. Lindsay Robert Wilson,
IJ Instruments Ltd.,
73 Harburn Road,
West Calder,
West Lothian,
EH55 8AT,
UK.

Web: http://www.eyejayinstruments.com
Email: sales@eyejayinstruments.com

## 14 REVISION HISTORY

| | |
|---|---|
| v1.3 (13/12/12): | Added comment re: moving window. |
| v1.2 (19/11/12): | Added comment on safety timeout feature. Corrected labels on Fig. 9. |
| v1.1 (30/06/12): | Re-arranged software installation section |
| v1.0 (27/06/12): | Initial |